Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

CLAIMS

1. (Currently amended) A server-based, computer implemented method for detecting and neutralizing ~~eliminating~~ invalid server-supplied data received from clients ~~machines~~ comprising the following steps performed following a server's [[the]] receipt of a request for services from a client ~~web browser~~ which request is accompanied by at least one identifier and associated ~~server~~ data placed on the client ~~web machine~~ via commands ~~for the web browser included in transport protocol response headers~~ sent to the client by the server or by related servers on earlier occasions, said method comprising:

scanning the ~~server~~ data which is received from the client ~~web browser~~ to identify, as invalid data, any data that contains improper characters;

determining the [[an]] identifier associated with ~~that accompanies~~ any data which is invalid; and

as part of a server response sent back to the client, ~~web browser,~~ including in the response a command ~~or commands~~ that causes only the invalid data, meaning character strings that include improper characters, identified by the identifier [[,]] to be neutralized.

2. (Currently amended) A method in accordance with claim 1, wherein the method is applied to the detection and neutralization of one or more cookies each associated with data and an identifier and supplied by the server or by related servers to clients, said method further comprising: ~~web browsers and,~~

when the [[its]] data and ~~name~~ the identifier associated with such a cookie is later returned by a particular client ~~web browser~~ to the server [[,]] and the data is found to contain invalid data, then neutralizing ~~and wherein only~~ cookies associated with ~~containing~~ invalid data [[,]] and identified by the associated identifier. ~~name, are neutralized.~~

3. (Currently amended) A method in accordance with claim 1, wherein ~~the server data accompanying~~ a request for services received from a client ~~web browser contains one~~ is accompanied by two or more separate identifiers ~~sets of data each including a name~~ and [[a]] associated data, ~~value,~~ and wherein the command or commands sent to the client as part of a response to the client includes one or more commands each of which identifies by identifier the associated ~~name a set of~~ data that contains invalid data which ~~and that~~ is to be neutralized, whereby other ~~sets of~~ data associated with other identifiers and containing valid data are not neutralized.

4. (Currently amended) A method in accordance with claim 1 [[3]], wherein neutralization is carried out by sending to a client a command that places on the client [[a]] new data [[set]] associated with an identifier found on the client associated with ~~a name for a~~ data [[set]] containing invalid data and a domain identifier of the server or of [[the]] related servers, the new data [[set]] containing a null data string ~~no erroneous data~~, whereby the new data [[set]] displaces the erroneous data [[set]] and thereby neutralizes the erroneous data [[set]].

5. (Currently amended) A method in accordance with claim 1, wherein ~~server~~ data placed on the [[a]] client ~~machine~~ via commands sent to the [[a]] client ~~web browser~~ includes an expiration date, and wherein neutralization is accomplished by adjusting the expiration date to a value ~~valve~~ that neutralizes the invalid data through expiration shortly after the commands are received.

6. (Original) A method in accordance with claim 5, wherein the expiration date is set to zero.

7. (Currently amended) A method in accordance with claim 5, wherein the expiration date is set to a date equal to or earlier than the date when the one or more commands are [[is]] sent back to the client.

8. (Currently amended) A method in accordance with claim 1, wherein the invalid data comprises data all of whose characters ~~value~~ should correspond[[s]] to one or more printable characters ~~identification codes~~ but some of whose characters ~~which~~ match characters ~~codes~~ contained in a list of invalid characters. ~~codes.~~

9. (Currently amended) A method in accordance with claim 1, wherein the data transfer protocol is HTTP or an equivalent protocol, the data received comprises one or more data sets preceded by a "Cookie:" command or its equivalent [[,]] and separated by semicolons ~~semi-colons~~ or an ~~some other~~ equivalent separator and of the form "NAME =VALUE" or an ~~some~~ equivalent form, and wherein the neutralization of such data is achieved by returning one or more commands "Set-cookie:" or an [[its]] equivalent command, each such command including at least a first expression that may be followed by one or more semicolons or equivalent separators and additional expressions ~~, separated by semi-colons or some equivalent separator,~~ of the form "NAME=VALUE" or an [[its]] equivalent expression, where NAME is the identifier ~~name~~ associated with invalid data and VALUE is valid data which may be no data.

10. (Currently amended) A method in accordance with claim 9, in which the command "Set-cookie:" or its equivalent is also followed by an expression "domain=DOMAIN_NAME" or an [[its]] equivalent expression, where DOMAIN_NAME identifies the server or the ~~group of~~ related servers.

11. (Currently amended) A method in accordance with claim 10, in which the command "Set-cookie:" or its equivalent is also followed by an expression "expires=DATE" or an [[its]] equivalent expression, where DATE is a date value or its equivalent adjusted to neutralize the invalid data ~~values~~ at [[by]] the client. ~~web browser.~~

12. (Original) A computer program containing instructions enabling it to cause a server to carry out the method steps as in claim 1.

13. (Currently amended) A server-based system for detecting and neutralizing ~~eliminating~~ invalid server-supplied data received back from clients ~~web browsers~~ comprising:

a server designed to communicate over a network with clients;

~~a client message receiver and transmitter on the server that is arranged to receive and to process incoming client messages and to transmit return messages back to clients;~~

a scanner that scans at least some requests for services ~~messages~~ flowing into the server coming from clients over the network and including a detector that can detect incoming identifiers and associated ~~server~~ data returned to the server by the client and originally placed on ~~supplied to~~ the client ~~on earlier occasions~~ by the server or by [[a]] related servers on earlier occasions;

a data integrity tester that tests the validity ~~integrity~~ of such incoming ~~server~~ data by searching the data for improper characters; and

a message insertion command generator placed into operation when the data integrity tester identifies invalid data, meaning data containing improper characters, in such incoming server data that causes the server, ~~message receiver and transmitter,~~ when transmitting a return message back to a client from which invalid data was received, to include within the return message at least one ~~or more~~ command[[s]] that causes the client to neutralize[[d]] the invalid data, identified by the associated identifier, without neutralizing other valid data.

14. (Currently amended) A system in accordance with claim 13, wherein the system is used ~~applied~~ to [[the]] detect[[ion]] and neutralize ~~neutralization of~~ one or more cookies supplied by the server or related servers to clients on earlier occasions, said system further comprising: ~~web browsers~~ and [[,]]

when the ~~its~~ data and identifier associated with a cookie are ~~name is~~ later returned to the server by a particular client ~~web browser to the server,~~ and the data integrity tester identifies ~~is found to contain~~ invalid data, ~~and wherein only~~ then the at least one command sent back to the client by the message insertion command generator neutralizes cookies associated with such identifiers and ~~containing~~ invalid data. ~~, identified by name, are neutralized.~~

15. (Currently amended) A system in accordance with claim 13, wherein ~~the server data accompanying~~ a request for services received from a client is accompanied by two ~~contains one~~ or more separate identifiers ~~sets of data each including a name~~ and associated [[a]] data, ~~value,~~ and wherein, if data is found to be invalid, the command or commands sent to the client by the message insertion command generator as part of a response to the client include[[s]] at least one ~~or more~~ command[[s]] ~~each of~~ which identifies by identifier the associated ~~name a set of~~ data that contains the invalid data which ~~and that~~ is to be neutralized, whereby other ~~sets of~~ data associated with other identifiers and containing [[in]]valid data are not neutralized.

16. (Currently amended) A system in accordance with claim 13 [[15]], wherein neutralization is carried out by the message insertion command generator sending to a client a command that places on the client ~~machine a~~ new data [[set]] associated with an identifier found on the client associated with ~~a name for the~~ data [[set]] containing invalid data and a domain identifier of the server or of related servers, the new data [[set]] containing a null data string ~~no erroneous data~~, whereby the new data [[set]] displaces the erroneous data [[set]] and thereby neutralizes the erroneous data [[set]].

17. (Currently amended) A system in accordance with claim 13, wherein the ~~server~~ data placed on the [[a]] client includes an expiration date, and wherein neutralization is accomplished by commands that adjust[[ing]] the expiration date to a value ~~valve~~ that neutralizes the invalid data through expiration shortly after the commands are received.

18. (Original) A system in accordance with claim 17, wherein the expiration date is set to zero.

19. (Original) A system in accordance with claim 17, wherein the expiration date is set to a date equal to or earlier than the date when the one or more commands are sent back to the client.

20. (Currently amended) A system in accordance with claim 13, wherein the invalid data comprises data all of whose characters ~~values~~ should correspond[[s]] to one or more printable characters ~~identification codes~~ but some of whose characters ~~which~~ match characters ~~codes~~ contained in a list of invalid characters. ~~codes.~~

21. (Currently amended) A system in accordance with claim 13, wherein the data transfer protocol is HTTP or an equivalent protocol, the data received comprises one or more data sets preceded by "Cookie:" or an equivalent command and separated by semicolons or an equivalent separator and of the form "NAME =VALUE" or an equivalent form, and wherein the neutralization of such data is achieved by returning one or more [[the]] commands "Set-cookie:" or an equivalent command, each such command including at least a first expression that may be followed by one or more expressions separated by semicolons or an equivalent separator of the form "NAME=VALUE" or an equivalent form where NAME is the identifier ~~name~~ associated with invalid data and VALUE is valid data or no data.

22. (Currently amended) A system in accordance with claim 21, in which the command "Set-cookie:" or its equivalent is also followed by an expression "domain=DOMAIN_NAME" or an equivalent expression, where DOMAIN_NAME identifies the server or the [[group of]] related servers.

23. (Currently amended) A system in accordance with claim 22, in which the command "Set-cookie:" or its equivalent is also followed by an expression "expires=DATE" or an equivalent expression [[,]] where DATE is a date value or its equivalent adjusted to neutralize the invalid data ~~value~~ at the client.

24. (Currently amended) A system in accordance with claim 21, in which the command "Set-cookie:" or its equivalent is also followed by an expression "expires=DATE" or an equivalent expression where DATE is a date value or its equivalent adjusted to neutralize the invalid data ~~value~~ at [[by]] the client. ~~browser.~~